

A new shift strategy for the implicitly restarted generalized second-order Arnoldi method

GONG FangHui & SUN YuQuan*

*School of Mathematics and Systems Science, BeiHang University, Beijing 100191, China;
Email: fanghui.gong@buaa.edu.cn, sunyq@buaa.edu.cn*

Received June 24, 2015; accepted June 24, 2015

Abstract In this paper, a new shift strategy for the implicitly restarted generalized second-order Arnoldi (GSOAR) method is proposed. In implicitly restarted processes, we can get a k -step GSOAR decomposition from a m -step GSOAR decomposition by performing $p = m - k$ implicit shifted QR iterations. The problem of the implicitly restarted GSOAR is the mismatch between the number of shifts and the dimension of the subspace. There are $2p$ shifts for p QR iterations. We use the shifts to filter out the unwanted information in the current subspace; when more shifts are used, one obtains a better updated subspace. But, if we use more than p shifts, the structure of the GSOAR decomposition will be destroyed. We propose a novel method which can use all $2p$ candidates and preserve the special structure. The new method vastly enhances the overall efficiency of the algorithm. Numerical experiments illustrate the efficiency of every restart process.

Keywords QEP, GSOAR method, implicitly restart, shifts, Ritz vector

MSC(2010) 65F15, 65Y20

Citation: Gong F H, Sun Y Q. A new shift strategy for the implicitly restarted generalized second-order Arnoldi method (in Chinese). *Sci Sin Math*, 2017, 47: 116, doi: 10.1360/012016-22. You can see the Chinese version online: <http://engine.scichina.com/doi/10.1360/012016-22>

1 Introduction

The quadratic eigenvalue problem (QEP)

$$Q(\lambda)x = (\lambda^2 M + \lambda C + K)x = 0 \quad (1.1)$$

has various applications in the engineering fields, like dynamic analysis of acoustic systems, fluid mechanics and microelectronic mechanical systems [3, 12, 15]. There are two major classes of numerical methods to solve large QEPs. One is to linearize the QEP into an equivalent generalized eigenvalue problem (GEP) such as

$$\begin{bmatrix} -C & -K \\ I & 0 \end{bmatrix} \begin{bmatrix} \lambda x \\ x \end{bmatrix} = \lambda \begin{bmatrix} M & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \lambda x \\ x \end{bmatrix}. \quad (1.2)$$

Currently, the GEP techniques have been quite mature [1, 7]. But the linearization methods have suffered some disadvantages: losing the spectral property of original QEP and doubling the dimension of the original one. Another class of methods work on the QEP directly. Numerically stable projection methods can keep the spectral properties as well as the structure of original QEP.

*Corresponding author

The Krylov subspace method plays a significant role in numerical techniques for solving large-scale GEPs. Bai and Su developed the standard Krylov subspace to second-order and proposed the second-order Arnoldi (SOAR) method for large-scale sparse QEPs [2]. They propose a SOAR procedure that computes an orthonormal basis of a second-order Krylov subspace. The SOAR method then projects (1.1) onto this subspace and computes the Ritz pairs to approximate the desired eigenpairs of (1.1). The general convergence property of Rayleigh-Ritz has been demonstrated in [4] and the refined Rayleigh-Ritz orthonormal projection method also has been proposed.

On account of the limitation of high computational cost and undue storage requirement, explicit or implicit restarting is usually necessary when we do large-scale matrices calculating. Implicit restarting is a powerful and extensively used technique in eigenvalue problem calculations [13]. Although the SOAR method plays an important role in solving QEPs, we can not directly apply the implicit restarting scheme to the SOAR procedure due to the special structure of the initial vector [10, 11]. Based on Bai and Su's work, many researchers work specifically to implicitly restart the SOAR procedure. Otto proposed a generalized SOAR (GSOAR) method which substitutes general vector for the original special-structure starting vector. The generalized SOAR procedure can directly apply implicit restarting technique under the hypothesis of no deflation [11]. An explicit restarting general Krylov subspace method was put forward after making a similar modification by Zhou [16]. Huang brought forward the implicit restarting semi orthogonal generalized Arnoldi (SGA) method [5]. At the same time, other researchers are constantly improving the SOAR method in all aspects [9].

Among these modified restarting SOAR methods, they all encountered the problem that implicit restarting failed to work when shifts did not match up with the dimensions of subspace. From recent researches, it can be seen that the quantity and quality of shifts are of great significance [9, 11]. Jia and Sun explored more properties and features of the GSOAR and gave an efficient method to compute refined Ritz vectors. They proposed a refined GSOAR (RGSOAR) method. Then they advanced certain exact shifts and refined shifts for respective use within the implicitly restarted GSOAR and RGSOAR algorithms. The refined shifts are based on the refined Ritz vectors. They presented an efficient algorithm to compute the exact and refined shift candidates reliably. Unlike the implicitly restarted algorithms for the linear eigenvalue problem, both exact and refined shift candidates are more than the shifts allowed. So the authors showed how to reasonably select the desired shifts among them. In addition, they proposed an effective approach to cure deflation in implicit restarts, so that implicit restarting is useable unconditionally.

Based on Jia and Sun's work, we give a new shift strategy for the implicitly restarted GSOAR and RGSOAR methods which can use all shifts. In the implicit restarting, we use the unwanted approximate eigenvalues as shifts to filter out the corresponding eigenvector information from the current subspace. A general result is that with more shifts approximating the unwanted eigenvalues, we can generate a better updated subspace. For the QEPs, one eigenvector can correspond to two different eigenvalues. If the eigenvalue pairs corresponding to the same eigenvector are present in exact or refined shifts, this means some essential shifts will be abandoned. This can be avoided by using all of the shift candidates. Obviously this will destroy the original structure. We use a subtle transformation making the damaged structure back to the Hessenberg form. Due to all shifts, the unwanted information is excluded more thoroughly and we can get a better subspace in each restarting. Because the implicit restarting in [9] is useable unconditionally, our algorithm is useable inherently.

The rest of this paper is arranged as follows. In section 2, we introduce the SOAR method and GSOAR method, implicit restarting, selection of exact and refined shifts. In section 3, we present the new strategy: using all shifts for projection method to enhance the efficiency of the method. In section 4, several numerical experiments are presented to illustrate the efficiency of the new implicitly restarted GSOAR method.

2 The implicitly restarted GSOAR and RGSOAR methods

The SOAR method was proposed to solve the large-scale sparse QEPs based on the second-order Krylov subspace by Bai and Su [2].

Definition 2.1. Let matrices $A, B \in \mathcal{C}^{n \times n}$, non-zero vector $u \in \mathcal{C}^n$, and define the sequence $r_0, r_1, r_2, \dots, r_{n-1}$ based on A, B and u , where

$$\begin{aligned} r_0 &= u \\ r_1 &= Au \\ r_j &= Ar_{j-1} + Br_{j-2} \quad \text{for } j \geq 2. \end{aligned}$$

The space

$$\mathcal{G}_n(A, B; u) = \text{span}\{r_0, r_1, r_2, \dots, r_{n-1}\} \quad (2.1)$$

is called a n th second-order Krylov subspace.

If we apply standard Krylov subspace technique to (1.2), then the corresponding Krylov subspace is:

$$\kappa_n(H, \nu) = \text{span}\{\nu, H\nu, H^2\nu, \dots, H^{n-1}\nu\}, \quad (2.2)$$

with $\nu = \begin{bmatrix} u \\ 0 \end{bmatrix}$. After some simple calculations, we note that there are some associations between the standard Krylov vectors $H^j\nu$ of length $2n$ in (2.2)

$$H^j\nu = \begin{bmatrix} r_j \\ r_{j-1} \end{bmatrix}, j \geq 1, \quad (2.3)$$

and the vectors of second-order Krylov subspace

$$r_j = Ar_{j-1} + Br_{j-2}. \quad (2.4)$$

The subspace (2.2) based on matrix H and starting vector v can be generalized by vector sequence $r_0, r_1, r_2, \dots, r_{n-1}$. In the meantime, (2.3) indicates that the second-order Krylov subspace $\mathcal{G}_j(A, B; u)$ can be used as the projection subspace of QEPs while we only need to generate an orthonormal basis $q_{j=1}^n$:

$$\mathcal{G}_j(A, B; u) = \text{span}\{q_1, q_2, \dots, q_j\}, j \geq 1. \quad (2.5)$$

Bai and Su presented a procedure for generating an orthonormal basis of the second-order Krylov subspace $\mathcal{G}_j(A, B; u)$.

Algorithm 1. SOAR procedure with deflation remedy

- 1: $q_1 = u_1 / \|u_1\|_2$
- 2: $p_1 = 0$
- 3: for $j = 1 : m$
- 4: $r = Aq_j + Bp_j$
- 5: $s = q_j$
- 6: for $i = 1 : j$
- 7: $t_{ij} = q_i^T r$
- 8: $r = r - q_i t_{ij}$

```

9:       $s = s - p_i t_{ij}$ 
10:    end for
11:     $t_{j+1,j} = \|r\|_2$ 
12:    if  $t_{j+1,j} = 0$ ,
13:      if  $s \in \text{span}\{p_i \mid i : q_i = 0, i \leq j\}$ 
14:        break
15:      else deflation
16:         $t_{j+1,j} = 1$ 
17:         $q_{j+1} = 0$ 
18:         $p_{j+1} = s$ 
19:      end if
20:    else
21:       $p_{j+1} = s/t_{j+1,j}$ 
22:       $q_{j+1} = r/t_{j+1,j}$ 
23:    end if
24:  end for

```

At step j , if r_i , $i = 0, 1, \dots, j$ are linearly dependent but $[r_i^T, r_{i-1}^T]^T$, $i = 0, \dots, j$ with $r_{-1} = 0$ are not, we call this situation deflation; if both $\{r_i\}$ and $\{[r_i^T, r_{i-1}^T]^T\}$ are linearly dependent at step j , we call this situation breakdown. If $t_{j+1,j} = 0$ at a certain step j ($j < k$), that means deflation occurs. In this case the second-order Krylov subspace $\mathcal{G}_j(A, B; u)$ does not contain any exact eigenvector of (1.1) [9]. Therefore, the measure to remedy deflation is to reset $t_{j+1,j}$ a non-zero value, here simply set it to be one, let $q_{j+1} = 0$, and continue the algorithm. The non-zero vectors of sequence $\{q_j\}$ still maintain orthonormal and span the subspace $\mathcal{G}_j(A, B; u)$.

For the quadratic eigenvalue problem (1.1) of n dimension, it corresponds to a second-order Krylov subspace $\mathcal{G}_m(A, B; u)$, with $m \leq n$, and $A = -M^{-1}C$, $B = -M^{-1}K$. Let $Q_m = [q_1, q_2, \dots, q_m] \in \mathcal{C}^{n \times m}$, $P_m = [p_1, p_2, \dots, p_m] \in \mathcal{C}^{n \times m}$ and unreduced upper Hessenberg matrix $\hat{T}_m = [t_{ij}] = \begin{bmatrix} T_m & \\ & t_{m+1,m} e_m^T \end{bmatrix} \in \mathcal{C}^{(m+1) \times m}$. If Algorithm 1 does not stop before m , then we have

$$\text{span}\{Q_m\} = \mathcal{G}_m(A, B; u), \quad (2.6)$$

and define

$$H \begin{bmatrix} Q_m \\ P_m \end{bmatrix} = \begin{bmatrix} Q_{m+1} \\ P_{m+1} \end{bmatrix} \hat{T}_m \quad (2.7)$$

as m -step SOAR decomposition, where $Q_{m+1} = [Q_m, q_{m+1}]$, $P_{m+1} = [P_m, p_{m+1}]$, $H = \begin{bmatrix} A & B \\ I & 0 \end{bmatrix}$.

Rayleigh-Ritz method can be adapted in (1.1) to seek the approximate eigenpairs (θ, y) satisfying the Galerkin restriction:

$$(\theta^2 M + \theta C + K)y \perp \mathcal{G}_m(A, B; u), \quad (2.8)$$

where $y \in \mathcal{G}_m(A, B; u)$, $\theta \in \mathcal{C}$. The columns of $n \times n$ matrix Q_m generated by Algorithm 1 are an orthonormal basis of $\mathcal{G}_m(A, B; u)$. Let θ, g satisfy the small-scale QEP:

$$(\theta^2 M_m + \theta C_m + K_m)g = 0, \quad (2.9)$$

where $M_m = Q_m^T M Q_m$, $C_m = Q_m^T C Q_m$, $K_m = Q_m^T K Q_m$, then we can obtain the approximate eigenpairs θ and $y = Q_m g$ of (1.1). The small-scale QEP (2.9) is produced by projecting the original QEP (1.1) onto $\mathcal{G}_m(A, B; u)$ explicitly. This method is named second-order Arnoldi (SOAR) method since it is an Arnoldi-like procedure. From the theory of Arnoldi method, we can find that the SOAR method has the advantages of fast convergence rate and simultaneous convergence of a group of eigenvalues. Moreover, it can still keep the special structure of original QEP after projection.

A disadvantage of this method is that implicit restarting scheme cannot be directly adapted. The new starting vector p_1 will no longer be zero after truncating. In order to satisfy the implicitly restarted conditions, researchers suggested the GSOAR method which replaces $p_1 = 0$ by $p_1 = u_2 / \|u_2\|_2$ and meets the requirement that starting vector $p_1 = u_2 / \|u_2\|_2$ is non-zero, where u_2 is a non-zero vector. Combining the deflation remedy SOAR procedure with GSOAR procedure, we get the following algorithm:

Algorithm 2. GSOAR procedure with deflation remedy

```

1:  $q_1 = u_1 / \|u_1\|_2, p_1 = u_2 / \|u_2\|_2$ .
2: for  $j = 1, 2, \dots, k$  do
3:    $r = Aq_j + Bp_j$ 
4:    $s = q_j$ 
5:   for  $i = 1, 2, \dots, j$  do
6:      $t_{ij} = q_i^* r$ 
7:      $r = r - t_{ij} q_i$ 
8:      $s = s - t_{ij} p_i$ 
9:   end for
10:   $t_{j+1j} = \|r\|$ 
11:  if  $t_{j+1j} = 0$ 
12:    if  $s \in \text{span}\{p_i | i : q_i = 0, 1 \leq i \leq j\}$ 
13:      break
14:    else deflation
15:      reset  $t_{j+1j} = 1$ 
16:       $q_{j+1} = 0$ 
17:       $p_{j+1} = s$ 
18:    end if
19:  else
20:     $q_{j+1} = r / t_{j+1j}$ 
21:     $p_{j+1} = s / t_{j+1j}$ 

```

22: end if

23: end for

If Algorithm 2 does not stop before m , it gives the following generalized second-order Krylov sequence and subspace; see [9].

Let A and B be $n \times n$ matrices and for vectors $u_1, u_2 \in \mathcal{C}^n$, and define

$$\begin{aligned} r_0 &= u_1, \\ r_1 &= Ar_0 + Bu_2, \\ r_j &= Ar_{j-1} + Br_{j-2} \quad \text{for } j \geq 2. \end{aligned}$$

Then $r_0, r_1, r_2, \dots, r_{m-1}$ is called a generalized second-order Krylov sequence based on A, B and u_1, u_2 , and $\mathcal{G}_m(A, B; u_1, u_2) = \text{span}\{r_0, r_1, r_2, \dots, r_{m-1}\}$ the m -th generalized second-order Krylov subspace.

Let $Q_m = [q_1, q_2, \dots, q_m] \in \mathcal{C}^{n \times m}$, $P_m = [p_1, p_2, \dots, p_m] \in \mathcal{C}^{n \times m}$ and $\hat{T}_m = [t_{ij}] = \begin{bmatrix} T_m \\ t_{m+1,m} e_m^T \end{bmatrix} \in \mathcal{C}^{(m+1) \times m}$. Then Q_m is an orthonormal basis of the m -th generalized second-order Krylov subspace $\mathcal{G}_m(A, B; u_1, u_2)$ and we get the m -step GSOAR decomposition:

$$H \begin{bmatrix} Q_m \\ P_m \end{bmatrix} = \begin{bmatrix} Q_{m+1} \\ P_{m+1} \end{bmatrix} \hat{T}_m, \quad (2.10)$$

where $Q_{m+1} = [Q_m, q_{m+1}]$, $P_{m+1} = [P_m, p_{m+1}]$, $H = \begin{bmatrix} A & B \\ I & 0 \end{bmatrix}$.

We can project the original QEP (1.1) onto the generalized second-order Krylov subspace $\mathcal{G}_m(A, B; u_1, u_2)$ explicitly. We still get the small-scale QEP:

$$(\theta^2 M_m + \theta C_m + K_m)g = 0, \quad (2.11)$$

where $M_m = Q_m^T M Q_m$, $C_m = Q_m^T C Q_m$, $K_m = Q_m^T K Q_m$. We can get the Ritz pairs θ and $y = Q_m g$ of (1.1) by computing the eigenpairs of (2.11). Then we get the GSOAR method. Suppose that we have computed the Ritz values θ by the GSOAR method. For each θ , we seek a unit length vector $\tilde{u} \in \mathcal{G}_m(A, B; u_1, u_2)$ satisfying the optimal requirement

$$\|(\theta^2 M + \theta C + K)\tilde{u}\| = \arg \min_{\substack{u \in \mathcal{G}_m(A, B; u_1, u_2) \\ \|u\| = 1}} \|(\theta^2 M + \theta C + K)u\| \quad (2.12)$$

and use it as an approximate eigenvector, called the refined Ritz vector. The pairs (θ, \tilde{u}) are also called the refined Rayleigh–Ritz approximations and the method is called refined generalized SOAR (RGSOAR) method. The refined vector can be computed by seeking a unit length vector $\tilde{z} \in \mathcal{C}^m$ such that $\tilde{u} = Q_m \tilde{z}$ with

$$\tilde{z} = \arg \min_{\substack{z \in \mathcal{C}^m \\ \|z\| = 1}} \|(\theta^2 M + \theta C + K)Q_m z\|, \quad (2.13)$$

the right singular vector of the matrix $(\theta^2 M Q_m + \theta C Q_m + K Q_m)$ associated with its smallest singular value $\sigma_{\min}(\theta^2 M Q_m + \theta C Q_m + K Q_m)$. It was shown in [9] that the singular vectors can be computed with moderate cost.

2.1 Implicit restarting

For the SOAR method and GSOAR method, in order to guarantee the convergence of the approximate eigenpairs, m should be as large as possible theoretically. Nevertheless, the storage and computing

requirements become massive as the dimension m of the subspace increases. So restarting is necessary for a practical method. If no deflation occurs, it is direct to adapt the implicit restarting scheme [13] to the GSOAR procedure. Given p shifts $\mu_1, \mu_2, \dots, \mu_p$, we can apply p implicit shifted QR algorithm to matrix T_m in (2.10)

$$(T_m - \mu_1 I) \cdots (T_m - \mu_p I) = V_m R, \quad (2.14)$$

where V_m is a $m \times m$ orthogonal matrix and R is a $m \times m$ upper triangular matrix. By relation (2.10), we have

$$H \begin{bmatrix} Q_m \\ P_m \end{bmatrix} V_m = \begin{bmatrix} Q_m \\ P_m \end{bmatrix} V_m (V_m^T T_m V_m) + t_{m+1,m} \begin{bmatrix} q_{m+1} \\ p_{m+1} \end{bmatrix} e_m^T V_m. \quad (2.15)$$

With simple computation, it is noted that $V_m^T T_m V_m$ is a Hessenberg matrix. If we take $\begin{bmatrix} \tilde{Q}_m \\ \tilde{P}_m \end{bmatrix} = \begin{bmatrix} Q_m \\ P_m \end{bmatrix} V_m$ and take $V_m^T T_m V_m = \tilde{T}_m$, then we have

$$H \begin{bmatrix} \tilde{Q}_m \\ \tilde{P}_m \end{bmatrix} = \begin{bmatrix} \tilde{Q}_m \\ \tilde{P}_m \end{bmatrix} \tilde{T}_m + t_{m+1,m} \begin{bmatrix} q_{m+1} \\ p_{m+1} \end{bmatrix} b_m^T, \quad (2.16)$$

where $b_m^T = (0 \ \cdots \ 0 \ b_{m-p} \ \cdots b_{m-1} \ b_m)$ which contains at least $m - p - 1$ zeros.

Given a m -step GSOAR decomposition (2.16), it can be truncated as a k -step GSOAR decomposition while $k \leq m - p$

$$H \begin{bmatrix} \tilde{Q}_k \\ \tilde{P}_k \end{bmatrix} = \begin{bmatrix} \tilde{Q}_k \\ \tilde{P}_k \end{bmatrix} \tilde{T}_k + t_{k+1,k} \begin{bmatrix} \tilde{q}_{k+1} \\ \tilde{p}_{k+1} \end{bmatrix} e_k^T. \quad (2.17)$$

Meanwhile, we get a k -th restarted subspace $G_k(A, B; u_1, u_2)$ spanned by an orthonormal vectors sequence $\{\tilde{q}_1, \tilde{q}_2, \dots, \tilde{q}_k\}$.

Because the implicitly restarted GSOAR method has a strict requirement for the dimension of restarted subspace, we can truncate (2.16) into a new k -th restarted subspace if and only if the subspace dimension $k \leq m - p$. Specifically, for (2.17), when k does not exceed $m - p$, the last term in the right looks like e_k^T after truncating and (2.17) will have the same form as (2.10). The k -step GSOAR decomposition can be expanded to m -step GSOAR decomposition again via Algorithm 2. Then looking for new Ritz pairs, computing their deviation and the process repeats.

Equation (2.17) is equivalent to a k -step GSOAR procedure mathematically. If deflations occur at steps $m_1, m_2, \dots, m_j \leq k$, then the corresponding j columns \tilde{q}_{m_j} of \tilde{Q}_k are zeros. Jia and Sun gave a method to cure this problem, so the implicit restarting is unconditional??.

2.2 Exact shifts and refined shifts

We consider the computation of shifts for GSOAR and RGSOAR methods in this subsection. Two kinds of shifts were given in [9]: the exact shifts and refined shifts. We first show how to compute the refined shifts as example.

The RGSOAR method computes the refined Ritz vectors \tilde{u}_i , which can be much more accurate than the Ritz vectors y_i . There were certain refined shifts for the refined Arnoldi method and the refined harmonic Arnoldi method for the linear eigenvalue problem [7, 8]. In the same spirit, the refined shifts were proposed for the RGSOAR method.

Let $\text{span}\{\hat{u}_1, \dots, \hat{u}_p\}$ be the orthogonal complement of the refined Ritz vectors \tilde{u}_i , $i = 1, 2, \dots, k$ with respect to $\mathcal{G}_k(A, B; q_1, p_1)$. Project QEP (1.1) onto the $\text{span}\{\hat{u}_1, \dots, \hat{u}_p\}$. Then we obtain a $p = m - k$ -dimensional projected QEP and compute its $2p$ eigenvalues. As a result, the $2p$ eigenvalues are the unwanted approximate eigenvalues. We call them refined shift candidates. We select $p = m - k$ refined shift candidates as shifts, called the refined shifts, for use within the implicitly restarted RGSOAR algorithm.

For GSOAR method, we can take the $2(m - k)$ unwanted Ritz values as the shift candidates directly. But it is just possible that one wanted Ritz value and one unwanted Ritz value correspond to the same Ritz vector. When we take the unwanted Ritz value as shift, that will filter out the wanted Ritz vector by restarting. To avoid this situation, we can compute the exact shift candidates in the orthogonal complement of the Ritz vectors $y_i, i = 1, 2, \dots, k$ with respect to $\mathcal{G}_k(A, B; q_1, p_1)$. We select $p = m - k$ exact shift candidates as shifts, called the exact shifts, for GSOAR method.

Both methods have $2p$ shift candidates, but if we use more than p shifts, equation (2.16) can not be truncated at the first k columns. So we can select the p shifts farthest from the target as shifts.

3 Shift strategy for implicitly restarted GSOAR and RGSOAR methods

In both methods, all refined vectors $\tilde{u}_1, \dots, \tilde{u}_k$ and Ritz vectors y_1, \dots, y_k are approximating the desired eigenvectors x_1, \dots, x_k . The exact and refined shift candidates are approximations to some of the unwanted eigenvalues of QEP (1.1) because the information on x_{k+1}, \dots, x_m has been removed from $\mathcal{G}_m(A, B; q_1, p_1)$. After restarting in (2.17), RGSOAR obtains $\text{span}\{\tilde{Q}_k\} = \text{span}\{\tilde{u}_1, \dots, \tilde{u}_k\}$ and GSOAR obtains $\text{span}\{\tilde{Q}'_k\} = \text{span}\{y_1, \dots, y_k\}$. Both methods achieve this goal through removing the unwanted information by shifts. As mentioned above, if two different shifts corresponding to same unwanted eigenvalue, at least one unwanted eigenvector can not be removed from the subspace $\mathcal{G}_m(A, B; q_1, p_1)$. The result of restarting can not be guaranteed. In order to achieve the restarting goal, all shift candidates must be used. The relationship between the number of shifts and the number of zero elements of $b_m^T = (0 \ \cdots \ 0 \ b_{m-p} \ \cdots b_{m-1} \ b_m)$ in (2.16) is shown in Figure 1. If the number of shifts p is more than $m - k$, the first k columns of (2.16) will not be a GSOAR decomposition. Because the last term in the right side is not only with e_k^T but also with e_{k-1}^T, e_{k-2}^T until e_{m-p}^T . So the implicit restarting technique can not be applied directly. Our main work is to propose an implicitly restarted generalized second-order Arnoldi strategy which can use all shift candidates.

It was shown in [14], that Krylov decomposition and Arnoldi decomposition are equivalent. If we can take Figure 1 (d) as a generalized Krylov decomposition, then it can be transformed to a GSOAR decomposition by orthogonal transformation. The scheme is written as the following result.

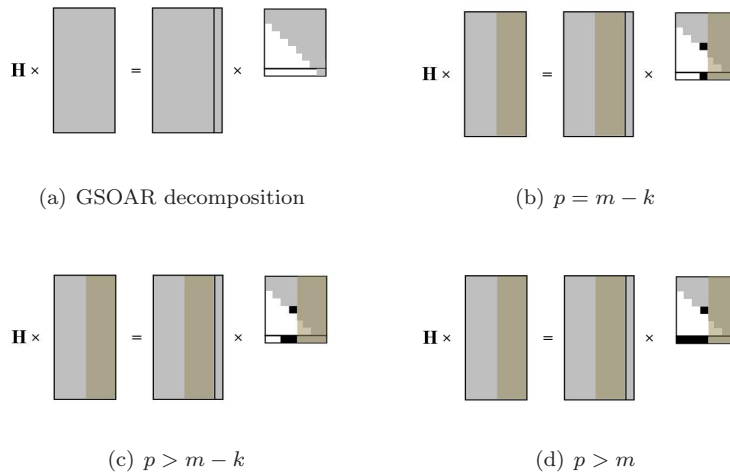


Figure 1 Implicitly restarted GSOAR method schematic

Theorem 3.1. For equation (2.16), there exists an orthonormal matrix W , that transforms b_m^T into αe_m^T , where α is a non-zero constant, and $W^T \tilde{T}_m W$ is a Hessenberg matrix.

Proof. Let W_1 be a Householder matrix, such that

$$W_1 b_m = \alpha e_m, \quad (3.1)$$

with α as a non-zero constant. Transpose of above equation,

$$b_m^T W_1^T = \alpha e_m^T, \quad (3.2)$$

with $W_1 = W_1^T$, we have

$$b_m^T W_1 = \alpha e_m^T. \quad (3.3)$$

Then multiply matrix W_1 on the right side of equation (2.16)

$$H \begin{bmatrix} \tilde{Q}_m \\ \tilde{P}_m \end{bmatrix} W_1 = \begin{bmatrix} \tilde{Q}_m \\ \tilde{P}_m \end{bmatrix} W_1 (W_1^T \tilde{T}_m W_1) + t_{m+1,m} \begin{bmatrix} q_{m+1} \\ p_{m+1} \end{bmatrix} b_m^T W_1. \quad (3.4)$$

Take $\begin{bmatrix} \tilde{Q}_m \\ \tilde{P}_m \end{bmatrix} W$ as a new $\begin{bmatrix} Q_m \\ P_m \end{bmatrix}$ and denote $W_1^T \tilde{T}_m W_1$ by new matrix B_m . Note that B_m loses the Hessenberg form. Combining the above two equations, we have

$$H \begin{bmatrix} Q_m \\ P_m \end{bmatrix} = \begin{bmatrix} Q_m \\ P_m \end{bmatrix} B_m + b_{m+1,m} \begin{bmatrix} q_{m+1} \\ p_{m+1} \end{bmatrix} e_m^T, \quad (3.5)$$

where $b_{m+1,m} = t_{m+1,m}\alpha$. At this moment, if we want to do implicit restarting, we must transform B_m into Hessenberg again and maintain the last term e_m^T .

Householder transformation and orthogonal transformation techniques are adopted once again to transform B_m into upper Hessenberg matrix. To prevent the next transformation damaging the previous result, the Householder transformations are performed on B_m from the bottom row to the top row.

When $l = 1$, denote B_m by matrix block form:

$$B_m = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1m} \\ b_{21} & b_{22} & \cdots & b_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ b_{m1} & b_{m2} & \cdots & b_{mm} \end{bmatrix} = \begin{bmatrix} B_{11} & B_{12} \\ c_1 & b_{mm} \end{bmatrix}, \quad (3.6)$$

where $c_1 = (b_{m,1} \ \cdots \ b_{m,m-1})$, $B_{11} = B_m(1 : m-1, 1 : m-1)$ is a $(m-1) \times (m-1)$ matrix, $B_{12} = B_m(1 : m-1, m)$ is a column vector of length $m-1$. Let R_1 be a $(m-1) \times (m-1)$ Householder matrix such that

$$c_1 R_1 = \alpha_1 e_{m-1}^T, \quad (3.7)$$

where α_1 is a non-zero constant. Let $U_1 = \begin{bmatrix} R_1 & 0 \\ 0 & 1 \end{bmatrix}$, we perform orthogonal transformation on B_m ,

$$B_m^{(1)} = U_1^T B_m U_1, \quad (3.8)$$

then $B_m^{(1)}$ has the following structure

$$U_1^T \begin{bmatrix} B_{11} & B_{12} \\ c_1 & b_{mm} \end{bmatrix} U_1 = \left[\begin{array}{c|c} B_{11}^{(1)} & B_{12}^{(1)} \\ \hline c_2 & \\ 0 & \cdots & 0 & B_{22}^{(1)} \end{array} \right], \quad (3.9)$$

where $B_{11}^{(1)}$ is a $(m-2) \times (m-2)$ matrix, $B_{12}^{(1)}$ is a $(m-2) \times 2$ matrix, $B_{22}^{(1)}$ is a 2×2 matrix and $c_2 = (b_{m-1,1} \ \cdots \ b_{m-1,m-2})$.

When $l = 2$, we can build a $(m - 2) \times (m - 2)$ Householder matrix R_2 , such that

$$c_2 R_2 = \alpha_2 e_{m-2}^T, \quad (3.10)$$

where α_2 is a non-zero constant and use $U_2 = \begin{bmatrix} R_2 & 0 \\ 0 & I_{2 \times 2} \end{bmatrix}$ to transform $B_m^{(1)}$ into

$$B_m^{(2)} = U_2^T B_m^{(1)} U_2 = \left[\begin{array}{c|c} \tilde{B}_{11}^{(1)} & \tilde{B}_{12}^{(1)} \\ \hline 0 & \cdots & \alpha_2 \\ 0 & \cdots & 0 \end{array} \middle| \begin{array}{c} B_{22}^{(1)} \end{array} \right] = \left[\begin{array}{c|c} B_{11}^{(2)} & B_{12}^{(2)} \\ \hline c_3 & \\ 0 & \cdots & 0 \\ 0 & \cdots & 0 \end{array} \middle| \begin{array}{c} B_{22}^{(2)} \end{array} \right], \quad (3.11)$$

where

$$B_{22}^{(2)} = \left[\begin{array}{c|c} * & *, * \\ \hline \alpha_2 & B_{22}^{(1)} \\ 0 & \end{array} \right] \quad (3.12)$$

is a 3×3 Hessenberg matrix.

Suppose $B_m^{(k-1)}$ is the following matrix block form:

$$B_m^{(k-1)} = \left[\begin{array}{c|c} B_{11}^{(k-1)} & B_{12}^{(k-1)} \\ \hline c_k & \\ 0 & \cdots & 0 \\ \vdots & \cdots & \vdots \\ 0 & \cdots & 0 \end{array} \middle| \begin{array}{c} B_{22}^{(k-1)} \end{array} \right], \quad (3.13)$$

where $B_{11}^{(k-1)}$ is a $(m - k) \times (m - k)$ matrix, $B_{12}^{(k-1)}$ is a $(m - k) \times k$ matrix, $B_{22}^{(k-1)}$ is a $k \times k$ Hessenberg matrix and $c_k = (b_{m-k+1,1} \cdots b_{m-k+1,m-k})$. We can build a $(m - k) \times (m - k)$ Householder matrix R_k , such that

$$c_k R_k = \alpha_k e_{m-k}^T, \quad (3.14)$$

where α_k is a non-zero constant. Let $U_k = \begin{bmatrix} R_k & 0 \\ 0 & I_{k \times k} \end{bmatrix}$, we obtain $B_m^{(k)}$:

$$B_m^{(k)} = U_k^T B_m^{(k-1)} U_k = \left[\begin{array}{c|c} \tilde{B}_{11}^{(k-1)} & \tilde{B}_{12}^{(k-1)} \\ \hline 0 & \cdots & \alpha_k \\ \vdots & \cdots & \vdots \\ 0 & \cdots & 0 \end{array} \middle| \begin{array}{c} B_{22}^{(k-1)} \end{array} \right] = \left[\begin{array}{c|c} B_{11}^{(k)} & B_{12}^{(k)} \\ \hline c_{k+1} & \\ 0 & \cdots & 0 \\ \vdots & \cdots & \vdots \\ 0 & \cdots & 0 \end{array} \middle| \begin{array}{c} B_{22}^{(k)} \end{array} \right], \quad (3.15)$$

with $B_{11}^{(k)}$ is a $(m - k - 1) \times (m - k - 1)$ matrix, $B_{12}^{(k)}$ is a $(m - k - 1) \times (k + 1)$ matrix, $B_{22}^{(k)}$ is a $(k + 1) \times (k + 1)$ Hessenberg matrix and $c_k = (b_{m-k,1} \cdots b_{m-k,m-k-1})$.

Obviously, when $l = m - 2$, $B_m^{(m-2)}$ is an upper Hessenberg matrix.

Let $U = U_1 U_2 \cdots U_{m-2}$, obviously U is an orthonormal matrix as U_1, U_2, \dots, U_{m-2} are orthonormal matrices. Multiply U on equation (3.5) from right:

$$H \begin{bmatrix} Q_m \\ P_m \end{bmatrix} U = \begin{bmatrix} Q_m \\ P_m \end{bmatrix} U (U^T B_m U) + b_{m+1,m} \begin{bmatrix} q_{m+1} \\ p_{m+1} \end{bmatrix} e_m^T U. \quad (3.16)$$

Because U_1, U_1, \dots, U_{m-2} have special structure, the last row of U must be e_m^T . Therefore, we have

$$e_m^T U = e_m^T. \quad (3.17)$$

We set $\begin{bmatrix} Q_m \\ P_m \end{bmatrix} U = \begin{bmatrix} Q'_m \\ P'_m \end{bmatrix}$ and $U^T B_m U = T'_m$, where T'_m is a new upper Hessenberg matrix. According to (2.10), (3.16) and (3.17), we find the orthonormal matrix $W = W_1 \times U$ and get the following equation

$$H \begin{bmatrix} Q'_m \\ P'_m \end{bmatrix} = \begin{bmatrix} Q'_m \\ P'_m \end{bmatrix} T'_m + b_{m+1,m} \begin{bmatrix} q_{m+1} \\ p_{m+1} \end{bmatrix} e_m^T. \quad (3.18)$$

Comparing (3.18) with (2.10), we complete the proof. \square

We can get the new k -step GSOAR decomposition by computing the first k columns of equation (3.18)

$$H \begin{bmatrix} Q'_k \\ P'_k \end{bmatrix} = \begin{bmatrix} Q'_k \\ P'_k \end{bmatrix} T'_k + t'_{k+1,k} \begin{bmatrix} q_{k+1} \\ p_{k+1} \end{bmatrix} e_k^T, \quad (3.19)$$

where the columns $\{q'_1, q'_2, \dots, q'_k\}$ of Q'_k are orthonormal vectors.

Algorithm 3. The implicitly restarted GSOAR method with all shifts

- 1: Run the m -step GSOAR procedure to generate orthonormal matrix Q_m , the columns of Q_m span the subspace $\mathcal{G}_m(A, B; u_1, u_2)$;
- 2: Do
 - a: Compute the wanted approximate eigenpairs;
 - b: If convergence then break, else:
 - c: Compute $2p$ exact or refined shifts;
 - d: Apply $2p$ implicit shifted QR algorithm to T_m ;
 - e: Compute W as described in Theorem 3.1 and get equation (3.18)

$$H \begin{bmatrix} Q'_m \\ P'_m \end{bmatrix} = \begin{bmatrix} Q'_m \\ P'_m \end{bmatrix} T'_m + b_{m+1,m} \begin{bmatrix} q_{m+1} \\ p_{m+1} \end{bmatrix} e_m^T;$$

- f: Truncate the above equation to get (3.19)

$$H \begin{bmatrix} Q'_k \\ P'_k \end{bmatrix} = \begin{bmatrix} Q'_k \\ P'_k \end{bmatrix} T'_k + t'_{k+1,k} \begin{bmatrix} q_{k+1} \\ p_{k+1} \end{bmatrix} e_k^T;$$

- g: Expand the k -step GSOAR decomposition to m .

End do

In Algorithm 3, in order to protect k desired eigenvalues, we usually preserve $k+l$ instead of k restarting vectors. Here, l is a small nonnegative integer, such as 3 [13]. One restarting in Algorithm 3 needs $m-1$ times Householder transformations and $3m-6$ times matrix multiplications using $O(m^3)$ flops totally more than traditional implicitly restarted methods. Nevertheless, the dimensions of operated vectors and matrices are reducing as the procedure runs. We use all the shift candidates to improve the efficiency of each restarting, which ultimately improves the overall efficiency.

4 Numerical experiments

Several numerical experiments are presented in this section to demonstrate the practicability and efficiency of Algorithm 3. We use IGSOAR to indicate the implicitly restarted second-order Arnoldi method with all exact shifts and IRGSOAR is the implicitly restarted second-order Arnoldi method with all refined shifts. We will show the superiority of the new methods. In addition, we will make a comparison between them and the corresponding counterparts IGSOAR0, IRGSOAR0 proposed in [9]. All the examples are run in the same environment: Win7, 64-bit operating system, Intel(R) Core(TM) i5-2430M CPU 2.4GHz, RAM 4GB using Matlab R2010a.

For all examples, nonnegative integer m stands for the dimension of projection subspace, k is the number of desired eigenvalues, f is the dimension of castoff subspace. So there are p shifts ($p = 2f$). In all tables, tol denotes the convergent criterion, TOTAL expresses the total CPU time. SOAR stands for the CPU time of computing the projected QEP and expanding GSOAR procedure from step $k + 1$ to step m , RESTART is the CPU time of performing implicit restarting and the time of solving projected QEP is denoted by FIND, unit is second. The abscissa axis is restart times and the vertical axis is the relative residual norms in following figures.

Example 4.1. This example is tested in [5, 9]. The 8010×8010 matrices are:

$$M = -4\pi^2 h^2 I_{q-1} \otimes (I_q - \frac{1}{2} e_q e_q^T), \quad C = 2\pi i \frac{h}{\xi} I_{q-1} \otimes (e_q e_q^T),$$

$$K = I_{q-1} \otimes D_q + T_{q-1} \otimes (-I_q + \frac{1}{2} e_q e_q^T),$$

where $h = 1/90$ is the mesh size, $q = 1/h$, \otimes is the Kronecker product, ξ stands for impedance, $D_q = \text{tridiag}(-1, 4, -1) - 2e_q e_q^T$, $T_{q-1} = \text{tridiag}(1, 0, 1)$. We adopt IRGSOAR, IGSOAR, IRGSOAR0 and IGSOAR0 to compute 6 desired approximate eigenvalues nearest to the origin with $m = 12$, $f = 5$, respectively. Table 1 and Figure 2 show the results.

Table 1 Example 4.1, $tol = 10^{-10}$

Algorithm	m	f	restarts	TOTAL	SOAR	RESTART	FIND
IRGSOAR	12	5	3	0.600	0.357	0.086	0.147
IGSOAR	12	5	3	0.521	0.324	0.076	0.109
IRGSOAR0	12	5	5	1.091	0.561	0.187	0.294
IGSOAR0	12	5	5	1.017	0.571	0.176	0.211

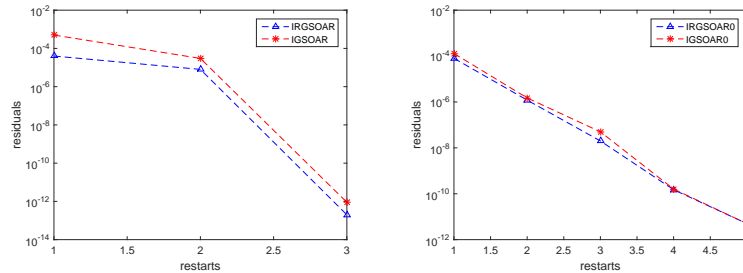
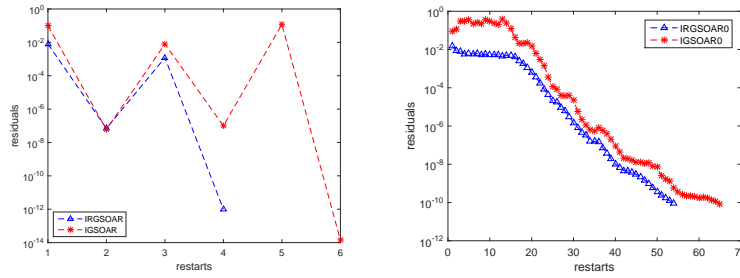


Figure 2 Example 4.1, $m = 12$, $f = 5$

It can be found from Table 1 and Figure 2 that IRGSOAR and IGSOAR work very well, their convergent speed is fast, moreover, total time is much less than IRGSOAR0 and IGSOAR0. New algorithms just need

Figure 3 Example 4.2.(a), $m = 40$, $f = 28$

3 restarts in contrast with 5 times of previous algorithms with part of shift candidates. The IRGSOAR and IGSOAR methods use 114 householder transformations with 0.012s. However, total time is saved much more than 0.012s. We notice that IRGSOAR is a little more efficient than the others.

Example 4.2. (a) This example is from [9] (cf. Example 4). The 5000×5000 matrices are:

$$M = I, \quad C = \tau \cdot \text{tridiag}(-1, 3, -1), \quad K = \kappa \cdot \text{tridiag}(-1, 3, -1),$$

where $\tau = 10$, $\kappa = 5$. We adopt IRGSOAR, IGSOAR, IRGSOAR0 and IGSOAR0 to compute 6 desired approximate eigenvalues nearest to target $\sigma = -13 + 0.4i$ with $m = 40$, $f = 28$, respectively. Table 2 and Figure 3 show the results.

Table 2 Example 4.2.(a), $\text{tol} = 10^{-10}$

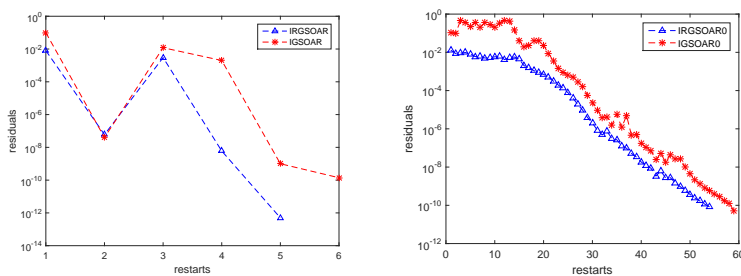
Algorithm	m	f	restarts	TOTAL	SOAR	RESTART	FIND
IRGSOAR	40	28	4	1.803	0.558	0.644	0.581
IGSOAR	40	28	6	2.567	0.700	1.174	0.673
IRGSOAR0	40	28	54	27.390	7.955	10.008	8.590
IGSOAR0	40	28	65	29.024	9.617	12.239	6.173

(b) If $f = 30$, other conditions are the same. Run the methods respectively. Table 3 and Figure 4 show the results.

Table 3 Example 4.2.(b), $\text{tol} = 10^{-10}$

Algorithm	m	f	restarts	TOTAL	SOAR	RESTART	FIND
IRGSOAR	40	30	5	2.341	0.616	0.979	0.730
IGSOAR	40	30	6	2.580	0.81	1.177	0.572
IRGSOAR0	40	30	54	26.880	8.154	10.013	7.877
IGSOAR0	40	30	59	25.508	8.514	10.665	5.416

We can see the marked superiority of IRGSOAR and IGSOAR algorithms. When $f = 28$, previous algorithms restarted more than 50 times to reach the convergent requirement. While IRGSOAR and IGSOAR algorithms just need 4 and 6 restarts, respectively. In addition, IRGSOAR saved almost 26s, in contrast to consuming 0.028s for Householder transformation. When $f = 30$, previous algorithms still needed more than 50 restarts to reach the convergent requirement. While IRGSOAR and IGSOAR

Figure 4 Example 4.2.(b), $m = 40$, $f = 30$

algorithms just need 5 and 6 restarts, respectively. IRGSOAR always need less restarts and less running time even when f changes. Example 2 tells us the implicit restarting methods are more advantageous in total running time and restart times from shifts increasing.

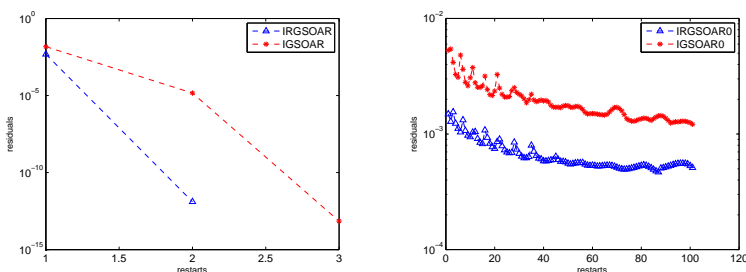
Example 4.3. (a) The size of this problem is $n = 5000$ and the matrices are

$$M = I_n, C = \begin{bmatrix} 8 & -4 & 0 & & \\ 2 & 12 & \ddots & & 0 \\ 0 & \ddots & 12 & -4 & \\ & 0 & 2 & 8 \end{bmatrix}, K = \begin{bmatrix} 2 & 2 & 0 & & \\ -1 & 3 & \ddots & & 0 \\ 0 & \ddots & 3 & 2 & \\ & 0 & -1 & 2 \end{bmatrix}.$$

We adopt IRGSOAR, IGSOAR, IRGSOAR0 and IGSOAR0 to compute 6 desired approximate eigenvalues nearest to target $\sigma = -10 - 0.8i$ with $m = 26$, $f = 15$, respectively. Table 4 and Figure 5 show the results.

Table 4 Example 4.3.(a), $tol = 10^{-10}$

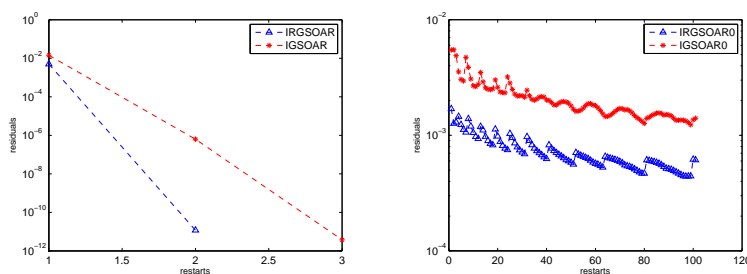
Algorithm	m	f	restarts	TOTAL	SOAR	RESTART	FIND
IRGSOAR	26	15	2	1.014	0.102	0.101	0.391
IGSOAR	26	15	3	0.936	0.194	0.175	0.129
IRGSOAR0	26	15	100	53.711	6.523	12.169	8.312
IGSOAR0	26	15	100	47.8455	6.494	12.095	5.497

Figure 5 Example 4.3.(a), $m = 26$, $f = 15$

(b) If $f = 13$, other conditions are the same. Run the methods respectively. Table 5 and Figure 6 show the results.

Table 5 Example 4.3.(b), $tol = 10^{-10}$

Algorithm	m	f	restarts	TOTAL	SOAR	RESTART	FIND
IRGSOAR	26	13	2	0.858	0.186	0.109	0.166
IGSOAR	26	13	3	0.858	0.186	0.171	0.158
IRGSOAR0	26	13	100	50.513	5.500	11.409	7.981
IGSOAR0	26	13	100	46.067	5.710	11.862	5.283

Figure 6 Example 4.3.(b), $m = 26$, $f = 13$

It can be seen from the results that the previous algorithms do not satisfy the convergent criterion yet after 100 restarts. However, the algorithms proposed in our work perform well on this problem. They only need two or three restarts to solve the large QEP.

We have developed implicitly restarted algorithms with certain exact and refined shifts proposed for generalized second-order Arnoldi method. Unlike the linear eigenvalue problem, for the QEP, there are more shift candidates than the shifts allowed. To overcome this problem, we have presented an efficient and reliable algorithm for using all shift candidates. Numerical experiments have demonstrated that using more shifts to improve each restart can boost efficiency significantly.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant No. 11201020).

References

- 1 Bai Z, Demmel J, Dongarra J, Ruhe A and van der Vorst H A. Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide. SIAM, Philadelphia, PA, 2000.
- 2 Bai Z and Su Y. SOAR: A second-order Arnoldi method for the solution of the quadratic eigenvalue problem. SIAM J. Matrix Anal, 2003, 26: 640–659.
- 3 Betcke T, Higham N J, Mehrmann V, Schröder C and Tisseur F. NLEVP: A collection of nonlinear eigenvalue problems. users' guide, MIMS EPrint, 2010.
- 4 Huang H M, Jia Z and Lin W W. On the convergence of Ritz pairs and refined Ritz vectors for quadratic eigenvalue problems. BIT Numerical Mathematics, 2013, 53: 941–958.
- 5 Huang W Q, Li T, Li Y T and Lin W W. A semiorthogonal generalized Arnoldi method and its variations for quadratic eigenvalue problems. Numerical Linear Algebra with Applications, 2013, 20: 259–280.
- 6 Jia Z. Refined iterative algorithms based on Arnoldi process for large unsymmetric eigenproblems. Linear Algebra Appl, 1997, 259: 1–23.
- 7 Jia Z. Polynomial characterizations of the approximate eigenvectors by the refined Arnoldi method and an implicitly restarted refined Arnoldi algorithm. Linear algebra and its applications, 1999, 287: 191–214.
- 8 Jia Z. The refined harmonic Arnoldi method and an implicitly restarted refined algorithm for computing interior eigenpairs of large matrices. Appl. Numer. Math, 2002, 42: 489–512.

- 9 Jia Z and Sun Y. Implicitly restarted generalized second-order Arnoldi type algorithms for the quadratic eigenvalue problem. *Taiwanese Journal of Mathematics*, 2015, 19: 1–30.
- 10 Meerbergen K. The quadratic Arnoldi method for the solution of the quadratic eigenvalue problem. *SIAM Journal on Matrix Analysis and Applications*, 2008, 30: 1463–1482.
- 11 Otto C. Arnoldi and JacobiCDavidson methods for quadratic eigenvalue problems. diploma thesis, Institut für Mathematik, Technische Universität Berlin, Germany, 2004.
- 12 Ruhe A. Algorithms for the nonlinear eigenvalue problem. *SIAM Journal on Numerical Analysis*, 1973, 10: 674–689.
- 13 Sorensen D C. Implicit application of polynomial filters in a k-step Arnoldi method. *SIAM Journal on Matrix Analysis and Applications*, 1992, 13: 357–385.
- 14 Stewart G W. A Krylov–Schur Algorithm for Large Eigenproblems. *SIAM Journal on Matrix Analysis and Applications*, 2002, 23: 601–614.
- 15 Tisseur F and Meerbergen K. The quadratic eigenvalue problem. *SIAM Rev*, 2001, 43: 235–286.
- 16 Zhou L, Bao L, Lin Y, Wei Y and Wu Q. Restarted generalized Krylov subspace methods for solving quadratic eigenvalue problems. *Inter. J. Comput. Math. Sci*, 2000, 4: 148–155.